

# Detector de las emociones que transmite la combinación cromática de una imagen

Jesús Orduña Heredia

**Resumen**—En este trabajo se ha desarrollado una aplicación de escritorio capaz de detectar que emociones transmite una imagen a partir de los colores que la componen. Estas emociones pueden ser sentimientos e ideas como por ejemplo alegría, afecto, diversión, atracción, salud, rústico, suave o ruido. Con el fin de cumplir este objetivo, se han transformado los colores de la imagen a diferentes espacios de color, para después implementar algoritmos de clustering que obtienen los colores más representativos de una imagen. Para representar sentimientos e ideas similares se han considerado emociones como *Cool*, *Earthy*, *Romantic* o *Serene*, representadas por un conjunto de colores, para la posterior clasificación de la imagen con la ayuda de algoritmos de clasificación. Para ello, se han estudiado los espacios RGB, CIELab y HSV, los algoritmos de clustering K-means y EM y los algoritmos de clasificación K-NN y SVM y, finalmente, se han comparado los resultados de las diferentes combinaciones de algoritmos. Después de seleccionar la combinación con mejores resultados, se ha diseñado una interfaz sencilla con la que poder seleccionar cualquier imagen y detectar las emociones que transmite.

**Palabras clave**—Detección de Emociones, Clasificación de Imágenes, Combinación de Colores, Procesamiento de Imágenes, Visión por Computador, Espacios de Color, Algoritmos de Clustering, Algoritmos de Clasificación.

**Abstract**—In this project, a desktop application able to detect the emotions that an image is able to transmit based on its colors has been developed. These emotions can be feelings or ideas such as joy, affection, fun, attraction, health, rustic, soft or noise. To achieve that, it has been necessary to transform the colors within the pictures to different color spaces, and then to apply clustering algorithms that can obtain the most representative colors of the image. To represent similar feelings and ideas, emotions like *Cool*, *Earthy*, *Romantic* or *Serene* have been considered, represented by a group of colors that are, afterwards, classified by the classification algorithms. To make it possible, the RGB, CIELab and HSV spaces have been studied, as well as the K-means and EM algorithms for clustering and the K-NN and SVM algorithms for classification. Finally, the results from the different combinations of methods have been compared, and, after selecting the best match among all the possibilities, a simple interface has been designed and implemented. With this interface, the user can select any picture and detect the emotions transmitted by it.

**Index Terms**—Emotion Detection, Image Classification, Color Combination, Image Processing, Computer Vision, Color Spaces, Clustering Algorithms, Classification Algorithm.

## 1 INTRODUCCIÓN

Vivimos en la era de la información. Cada día vemos una infinidad de imágenes tanto por la calle como en televisión, ordenadores o móviles, y todas ellas nos transmiten algún tipo de sentimiento. De hecho, lo que solemos ver en informativos, películas, series, publicidad o incluso en páginas web está editado con el fin de transmitir unas determinadas emociones (tranquilidad, alegría, tristeza, miedo, etc.). Para conseguir este objetivo, los diseñadores gráficos editan los objetos del contenido dándoles una forma, una distribución y unos colores para lograr el efecto que buscan.

Bajo este contexto, se han llevado a cabo numerosos estudios sobre qué es lo que nos impacta al ver una imagen, dando como resultado que la combinación de colores es, junto con la forma, la causa principal de los sentimientos que nos produce [1].

Cada vez son más las empresas que tienen en cuenta estas afirmaciones y buscan estrategias de marketing donde el color es un factor importante en la publicidad y la imagen de la empresa. Así pues, en cualquier anuncio podemos ver que, dependiendo del producto, las empresas utilizan diferentes combinaciones de colores para atraer a diferentes sectores sociales (p.e. el Grupo Lego utiliza colores muy vivos en sus productos ya que buscan un público infantil). Por este motivo, en estos últimos años se está investigando la relación entre la composición cromática y las emociones que puede llegar a transmitir una imagen.

Partiendo de esta base, se propone desarrollar una aplicación que sirva de ayuda en la investigación de la relación mencionada y en la de otros ámbitos como, por ejemplo, el de la publicidad donde se buscan combinaciones de colores en sus campañas para atraer el interés de un sector social concreto. Así pues, el proyecto se centrará en el desarrollo de una herramienta capaz de determinar las emociones que transmite una imagen seleccionada. La aplicación será un entorno sencillo e intuitivo donde se enmascarará una lógica algorítmica que se detallará a lo largo del proyecto.

- E-mail de contacto: [jesusorhe@gmail.com](mailto:jesusorhe@gmail.com)
- Menció n realizada: Ingeniería de Computación.
- Trabajo tutorizado por: Robert Benavente Vidal (Ciencias de la Computación)
- Curso 2014/15

## 2 OBJETIVOS

El principal objetivo del proyecto es crear una herramienta que permita al usuario saber las emociones (interés) que transmite una imagen concreta. La obtención de emociones se basará en los sentimientos (impresiones) que transmite la combinación cromática. Por lo tanto, el desarrollo principal del proyecto será la creación de un algoritmo capaz de detectar y clasificar la gama cromática de una imagen según unas emociones establecidas a partir de una serie de combinaciones de colores.

El cumplimiento de este objetivo principal implica el desarrollo de las siguientes tareas, las cuales plantean unos objetivos secundarios del proyecto:

- Obtener los píxeles de la imagen en diferentes espacios de color. Para ello se estudiarán las características y ventajas que proporcionan diferentes espacios de color aplicados en el procesamiento de imágenes.
- Obtener los colores predominantes de la imagen. Para ello se estudiará la funcionalidad y el rendimiento de los algoritmos de clasificación no supervisada más comunes en el procesamiento de imágenes para obtener los colores predominantes de una imagen.
- Obtener la emoción que transmite la imagen. Para ello se estudiará la funcionalidad y el rendimiento de los algoritmos de clasificación supervisada más comunes en el procesamiento de imágenes para clasificar los colores predominantes de una imagen a partir de una lista de emociones.
- Diseñar una interfaz de usuario sencilla e intuitiva para poder seleccionar imágenes y obtener las emociones que transmiten.

## 3 ESTADO DEL ARTE

Antes de empezar a plantear el funcionamiento de la aplicación, se ha hecho un trabajo previo buscando investigaciones y aplicaciones relacionadas con las emociones y los colores.

El primer trabajo que propuso un método para clasificar imágenes fue el de Solli et al. [2] que se centra en los valores *activity*, *weight* y *heat* para clasificar de forma genérica grupos de imágenes. Por otra parte, Shin et al. [3] investigaron sobre las emociones que transmiten unos determinados tejidos, basándose en conjuntos de texturas que, además del color, buscan patrones que pueda haber en las imágenes.

Aproximándose a la idea del detector de emociones, Chaitanya et al. [4] crea una relación entre la combinación de colores definidos en el libro "*Color Image Scale*" [5] y los histogramas de las imágenes para, finalmente, asociarlas a una serie de conceptos. Finalmente Csúrká [6] y Murray [7] cambian los colores de las imágenes, utilizando diferentes métodos, para que transmitan las emociones escogidas a partir de diferentes paletas de colores determinadas en el libro "*Pantone guide to communication with color*" [8].

Después de analizar la metodología utilizada en los artículos anteriores, se decidió utilizar el método propuesto

en [7], como base del proyecto desarrollado. En él, se detallan las pruebas realizadas con el algoritmo de clustering EM (Expectation – Maximization) + GMM (Gaussian Mixture Model) para obtener los colores, que mejor representen la imagen, las estructuras GMM y KDE (Kernel Density Estimation), para encontrar modelos de representación de datos, y el algoritmo de clasificación SVM (Support Vector Machine) para relacionar las gamas de colores a las emociones descritas en la referencia [8] (los conceptos EM, GMM, KDE y SVM están explicados en la referencia [9]).

## 4 SISTEMA DESARROLLADO

El funcionamiento de la lógica de la aplicación consta de los siguientes pasos:

1. A partir de conjuntos de colores asociados a emociones, se han creado modelos de cada emoción.
2. Dada una imagen, extraer los colores predominantes que mejor representen su combinación cromática. Para esto, se han estudiado y comparado diferentes espacios de color con los que tratar la imagen.
3. Para obtener los colores predominantes se utilizan algoritmos de clasificación no supervisada, más conocidos como algoritmos de clustering.
4. Se han clasificado los colores predominantes de la imagen con algoritmos de clasificación utilizando los modelos generados.
5. Con las clasificaciones de cada color predominante, se ha seleccionado la emoción final mediante un sistema de votación.

En la Figura 1 se puede ver un esquema del funcionamiento de la lógica del sistema.

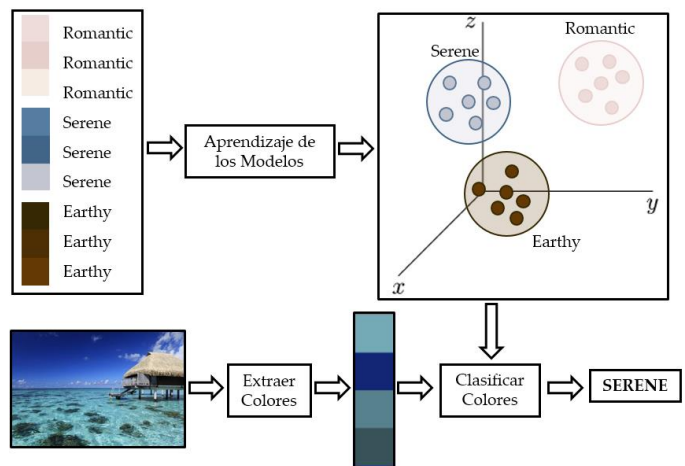


Figura 1. Diagrama de flujo de la lógica del detector.

Para la implementación del código se ha utilizado el lenguaje orientado a objetos Python 2.7 [10], la librería OpenCV 2.4 [11] para los algoritmos de clustering y de clasificación y NumPy 1.9 [12] para las estructuras de datos.

## 5 COLORES PREDOMINANTES DE LA IMAGEN

Para empezar a reconocer los colores de la imagen se debe tener en cuenta en qué espacio de color se representan los píxeles de la imagen, ya que cada espacio tiene sus propias características.

Por lo tanto, en los siguientes apartados se describirá cómo se ha llevado a cabo el estudio y la implementación de los espacios de color y de los algoritmos de clustering.

### 5.1. ESPACIOS DE COLOR

En el desarrollo del proyecto se han considerado los espacios de color RGB, CIELab y HSV [13], ya que son los más utilizados en el procesamiento de imágenes debido a las diferentes ventajas que proporciona cada uno.

Para su posterior utilización, la imagen se ha transformado al espacio seleccionado a partir del espacio RGB.

Por lo tanto el primer espacio en ser implementado ha sido RGB, porque es la representación más extendida en los formatos de imágenes. Esto se debe a que cada canal representa un color primario (Red, Green, Blue) que combinando sus valores generan el espacio de colores. Como cada canal representa un color, estos están definidos en el mismo rango de valores 0 - 255 creando un espacio tridimensional con forma de cubo, donde cada una de las esquinas representa el valor absoluto de los colores Negro, Rojo, Verde, Azul, Magenta, Amarillo, Cian y Blanco.

El siguiente espacio implementado ha sido CIELab que es el utilizado en la mayoría de trabajos sobre el tema de la relación de emociones con el color. Esto no es de extrañar si tenemos en cuenta que su propósito principal es el de crear un espacio de color que sea más perceptivamente uniforme. Esto significa que un cambio en los valores de color debe tener la misma relevancia que un cambio en la percepción visual. El espacio de valores de CIELab se representa con tres canales, uno para la intensidad (Lightness) con valores entre 0 y 100, y dos para definir la tonalidad (componentes A y B) con valores negativos y positivos.

El último espacio implementado ha sido HSV debido a que este da especial importancia a la saturación y la luminosidad de los colores y, una de las características de la combinación de colores aplicada a la emociones, es que los colores con saturaciones y luminosidades parecidos suelen transmitir emociones similares. Por lo tanto, los tres canales son el matiz del color (Hue) con valores angulares entre 0 y 360, la saturación (Saturation) con valores entre 0 y 100, y la intensidad (Value o Brightness) con valores entre 0 y 100.

### 5.2. ALGORITMOS DE CLUSTERING

Para poder analizar la gama cromática de la imagen se deben analizar los colores de todos los píxeles que la componen pero, cómo esto supondría un coste de ejecución muy elevado, se ha optado por tomar 9 valores que representen lo mejor posible el conjunto de colores de la imagen.

Para poder encontrar estos 9 colores predominantes, se han utilizado algoritmos de clustering. Estos buscan los K clusters (un cluster es una agrupación de puntos) que

mejor definan los colores de la imagen, donde K es un número definido de clusters que los algoritmos encuentran al cumplir ciertas condiciones y, finalmente, se selecciona un valor que represente a cada cluster.

Además, para que las ejecuciones de estos algoritmos no ocupen mucho tiempo de ejecución, se ha definido un tamaño de imagen máximo de 200 x 200 píxeles por lo que se redimensionarían las que superasen estas dimensiones.

Por lo tanto, una vez la imagen se ha cargado, se redimensiona si es necesario, se transforma al espacio de color seleccionado y se normalizan los datos para que los algoritmos de clustering puedan hacer los cálculos lo más precisos posible. En este caso, los datos se han normalizado entre 0 y 1 ya que es un requisito de la implementación utilizada de los algoritmos de clustering.

En este proyecto, se han implementado los algoritmos K-means y EM [9], por ser los más comunes dentro del marco de clustering.

#### 5.2.1. K-means

K-means es uno de los algoritmos de clustering más sencillos ya que únicamente se basa en la distancia euclidiana entre los puntos. Como se detalla en la Figura 2, a partir de un número K de centroides (un centroide es el punto central que representa a un grupo de puntos) generados aleatoriamente sobre el espacio de puntos proporcionado (en este caso los píxeles en el espacio seleccionado). Después, se asigna cada píxel a un centro a partir de la distancia euclidiana que hay hasta cada centroide y se escoge el más cercano.

El siguiente paso es actualizar la posición del centroide calculando la media de los puntos asignados. Finalmente, se repite el proceso hasta que los centroides no cambien.

```

Inicio
Inicializar  $n, K, c_i, c_1, \dots, c_K$ 
Hacer
    Clasificar  $n$  puntos en el  $c_i$  más cercano
    Recalcular  $c_i$ 
Hasta no hay cambio en  $c_i$ 
Retornar  $c_1, c_2, \dots, c_K$ 
Fin
  
```

Figura 2. Pseudocódigo de K-means donde  $n$  es el número de muestras o píxeles,  $c_i$  es el centroide  $i$  y  $K$  es el número de clusters.

En el K-means implementado los centroides son generados aleatoriamente, el valor de K está definido a 9 y se han establecido dos criterios de parada; el algoritmo finalizará bien cuando haya hecho 100 iteraciones o cuando el desplazamiento de los centroides sea menor a 0.001 %.

#### 5.2.2. EM

EM es un algoritmo de máxima verosimilitud, ya que le da más importancia a la cantidad de puntos en zonas del espacio que a la distancia que hay entre ellos. Normalmente, EM utiliza GMM para el cálculo de los valores que estiman la relación entre los puntos. Por lo tanto, GMM le proporciona a EM la capacidad de valorar la densidad de puntos a partir de la media y la desviación estándar de la Gaussiana.

De esta manera, el centro de la Gaussiana representa el valor promedio de los puntos que se encuentran dentro de ella. En la Figura 3 se detalla cómo EM tiene dos pasos principales, el paso de expectativa E y el paso de maximización M.

En el paso de expectativa, se utilizan aproximaciones iniciales por cada punto en las Gaussianas mediante el cálculo de los valores esperados  $\theta^i$ , donde  $\theta^i$  son la media y la desviación estándar de las Gaussianas.

En el paso de maximización, después de calcular los valores esperados, se calcula la siguiente componente  $\theta^{i+1}$  a partir de los valores de pertenencia de cada punto a su Gaussiana.

Una vez calculado el valor  $\theta^{i+1}$ , el paso de expectativa se repite para volver a calcular los nuevos valores esperados hasta que estos parámetros converjan.

**Inicio**

**Inicializar**  $\theta^{i+1}$ ,  $T$ ,  $i = 0$

**Hacer**

Paso E: ejecutar  $Q(\theta; \theta^i)$

Paso M:  $\theta^{i+1} = \max(Q(\theta; \theta^i))$

**Hasta** diferencia  $(\theta \rightarrow \theta^{i+1}) - \text{diferencia}(\theta^{i+1} \rightarrow \theta^i) \leq T$

**Retornar**  $\theta^{i+1}$

**Fin**

Figura 3. Pseudocódigo de EM donde  $\theta^i$  son los valores esperados,  $T$  es el porcentaje de convergencia,  $i$  es el número de iteraciones y la función  $Q()$  calcula los valores esperados.

En el EM implementado los valores se inicializan aleatoriamente y, al igual que el K-means, el valor de la K está definido a 9 y se han utilizado dos criterios de parada, aunque esta vez los valores son 10 para el número máximo de iteraciones y 0.1 % para la diferencia de los valores entre la iteración anterior y la nueva.

## 6 CLASIFICACIÓN DE LA IMAGEN

Para poder detectar las emociones que transmite una imagen, se debe poder analizar y relacionar su gama cromática con algún tipo de patrón que defina una relación entre combinaciones de colores y emociones.

Así pues, los colores predominantes de la imagen se pueden correlacionar con una gama de colores asociados a emociones mediante algoritmos de clasificación supervisada.

### 6.1. COLORES Y EMOCIONES

En alguno de los trabajos expuestos en el “Estado del Arte”, se han basado en los libros [5] y [8] para crear los clasificadores. Para escoger la mejor relación entre colores y emociones se han consultado dichos libros y, finalmente, se ha decidido utilizar el segundo debido a que engloba conjuntos de sentimientos similares en unas pocas emociones. Los dos libros tienen en común que la combinación de tres colores es suficiente para transmitir un gran abanico de emociones, pero el seleccionado separa estos tres colores en principal, secundario y relevante, dependiendo del grado de importancia que tenga sobre la imagen. El color principal es el que está más presente en toda la imagen, el secundario el segundo más presente, y el relevante el que destaca sobre el resto.

Por lo tanto, a partir de la información proporcionada, se han escogido 8 de las emociones que se proponen (*Cool, Earthy, Playful, Romantic, Sensual, Serene, Spiritual y Warm*) cómo se puede ver en la Figura 4.



Figura 4. Distribución de colores de las emociones *Cool* (fila 1), *Earthy* (fila 2) y *Sensual* (fila 3) en los espacios RGB, CIELab y HSV.

Basado en el sistema de combinación de tres colores propuesto, se han creado tres diccionarios (el de los colores principales, el de secundarios y el de los relevantes). En cada uno, se relaciona cada color, representado en la escala Pantone [14], con las emociones que puede transmitir combinándolo con otros. Para cada una de las 8 emociones se han incluido 22 colores en cada uno de los diccionarios, haciendo un total de 176 colores en cada uno relacionados cómo se puede observar en (1).

$$D_{\text{principal}} = \{(p_0, e_0), (p_1, e_0), \dots, (p_i, e_i)\} \quad (1)$$

En (1) está representada la forma que tienen los diccionarios de colores principales, donde  $p$  es un color representado en la escala Pantone y  $e$  el nombre de una de las 8 emociones.

Además de tener estos diccionarios, también se ha generado un diccionario con todos los colores Pantone y sus correspondientes representaciones en los espacios Hexadecimal, RGB, CIELab y HSV, dando como resultado una representación de las emociones cómo la que se muestra en la Figura 5.

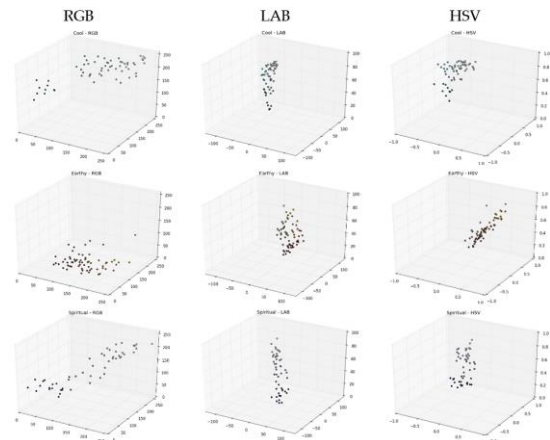


Figura 5. Distribución de colores de las emociones *Cool* (fila 1), *Earthy* (fila 2) y *Sensual* (fila 3) en los espacios RGB, CIELab y HSV.

## 6.2. ALGORITMOS DE CLASIFICACIÓN

Los algoritmos de clasificación supervisada son aquellos que requieren un entrenamiento a partir de un conjunto de datos ya clasificados para la clasificación de futuros valores. En el proyecto, el conjunto de entrenamiento se basa en los colores etiquetados con las emociones alojados en los diccionarios explicados en el apartado anterior y, los nuevos valores a clasificar son los colores predominantes de la imagen extraídos de los algoritmos de clustering.

Los algoritmos implementados en el proyecto han sido K-NN (K-Nearest Neighbors) y SVM [9] por ser de los más extendidos en el marco de Machine Learning y se adaptan fácilmente a los datos que se tratan en el proyecto.

### 6.2.1. K-NN

K-NN es uno de los algoritmos de clasificación supervisada más sencillos, ya que se centra en encontrar los K vecinos más cercanos de una serie de puntos a clasificar.

Para empezar, a K-NN se le debe proporcionar un conjunto de entrenamiento con puntos etiquetados previamente. Una vez entrenado, se debe indicar un valor K que define el número de vecinos cercanos que el algoritmo deberá obtener para cada uno de los nuevos valores a clasificar. Para determinar los vecinos, K-NN busca los puntos más cercanos calculando la distancia euclidiana.

En el caso del algoritmo implementado en el proyecto el valor K está definido a 9.

### 6.2.2. SVM

SVM es uno de los algoritmos de clasificación supervisada más utilizados debido a que proporciona diferentes maneras de buscar la mejor opción para un problema en concreto.

SVM se basa en la creación de hiperplanos N-dimensionales, donde N es el número de dimensiones de los datos, que separen de la mejor manera posible los diferentes conjuntos de entrenamiento. Al igual que con K-NN, estos conjuntos se deberán pasar como un parámetro con sus respectivas clasificaciones.

Para separar los conjuntos de entrenamiento, SVM busca el margen que maximice la distancia entre las fronteras de los conjuntos. Existen diferentes kernels (funciones) que utilizan hiperplanos para delimitar estos márgenes, aunque la fórmula genérica es (2).

$$g(y) = a\phi(x) + b \quad (2)$$

En la fórmula (2)  $\phi(x)$  es la función del kernel que mapea los puntos x en nuevas dimensiones.

Algunos de los kernels más utilizados en SVM son el lineal, el polinómico, el radial o RBF (Radius Basis Function) y el sigmoidal. Además, cada uno de estos kernels utiliza diferentes parámetros para poder definir correctamente sus funciones [15].

Cuándo SVM ya está entrenado y se han obtenido los parámetros que mejor separan las clases, ya está preparado para clasificar nuevos valores.

En este caso, la implementación de SVM en la librería

OpenCV proporciona herramientas para poder encontrar el mejor modelo de SVM para el caso específico en el que se esté trabajando, facilitando la implementación final en el detector de emociones.

En el algoritmo implementado se ha utilizado el kernel RBF con los parámetros predefinidos ya que suele ser el que mejores resultados obtiene en las clasificaciones multiclase.

## 6.3. MÉTODOS DE CLASIFICACIÓN Y VOTACIÓN

Cómo se ha comentado en el apartado anterior, los algoritmos de clasificación supervisada tienen dos pasos, entrenamiento y clasificación y, la manera de hacerlo puede ser diferente dependiendo del resultado que se busque. Por este motivo, se han ideado diferentes métodos de clasificación y diferentes tipos de votación.

### 6.3.1. Métodos de Clasificación

En el proyecto se han implementado dos métodos para el entrenamiento y la posterior clasificación, que se han llamado Simple y Complejo.

#### Método Simple

En este método se utilizan todos los colores de todas las emociones para crear un único conjunto de entrenamiento para la creación del modelo del clasificador y, a la hora de clasificar los colores de la imagen se clasifican todos los colores predominantes obtenidos.

#### Método Complejo

En este método se crea un conjunto de entrenamiento por cada diccionario de colores (uno para los principales, otro para los secundarios y otro para los relevantes) creando tres modelos del clasificador. Por lo tanto, a la hora de clasificar, se escogen los colores predominantes que representen mejor el color principal, el secundario y el relevante de la imagen y se clasifica cada uno con el modelo correspondiente.

### 6.3.2. Tipos de Votaciones

Una vez se han clasificado los colores predominantes con alguno de los métodos anteriores, se debe hacer una votación de los resultados para obtener una clasificación final que represente la emoción que transmite la imagen. Por ello, se ha implementado una votación global y una por valor.

#### Votación global

Este tipo de votación considera todas las clasificaciones obtenidas para todos los colores y se selecciona la que más se repite obteniendo, directamente, la emoción que transmite la imagen.

#### Votación por valor

Este tipo de votación, considera las clasificaciones obtenidas en cada color y se selecciona, por cada valor, la que más se repite. Después, se cogen las selecciones anteriores y se repite el proceso obteniendo, finalmente, la emoción que transmite la imagen.

## 7 RESULTADOS

Para determinar la eficacia de los métodos implementados, en los siguientes apartados se comparan los resultados de la combinación de todos ellos, los tiempos de ejecución y las clasificaciones de algunas imágenes.

### 7.1. OBTENCIÓN DE LOS MEJORES MÉTODOS

Durante el proyecto se han implementado y comparado las 48 posibles combinaciones de métodos:

- Espacios de color: RGB, CIELab, HSV.
- Algoritmos de clustering: K-means, EM.
- Algoritmos de clasificación: K-NN, SVM.
- Métodos de clasificación: Simple, Complejo.
- Métodos de votación: Valor, Global.

Para elegir la mejor combinación de métodos se han hecho pruebas clasificando un conjunto de test. Este conjunto se ha creado buscando las emociones y sus descripciones en Google Imágenes y está compuesto por un total de 80 imágenes, 10 de cada emoción. A continuación, se ha ejecutado cada combinación, clasificando todas las imágenes en una de las emociones clasificadas, hasta 10 veces, haciendo un total de 800 clasificaciones. Finalmente, se han comparado las clasificaciones reales con las obtenidas con la aplicación calculando la Accuracy (3) como muestra la Tabla 1.

$$\text{Accuracy} = \frac{\text{Resultados Correctos}}{\text{Resultados Correctos} + \text{Resultados Incorrectos}} \quad (3)$$

Analizando la Tabla 1 se puede observar que el clasificador K-NN, en general, obtiene mejores resultados que el SVM, seguramente porque mientras que el primero extrae 9 resultados, el segundo solamente 1. No obstante, para que el clasificador SVM mejorara sus resultados podría hacerse una búsqueda exhaustiva de parámetros.

Respecto al método de clasificación, los mejores resultados parece que se obtienen casi siempre con el Simple, tanto utilizando K-NN como SVM. Posiblemente esto se deba a que el método Simple utiliza más muestras en el entrenamiento y en la clasificación que con el método Complejo.

En cuanto a la votación final, utilizando SVM los resultados son muy similares entre ellos. Sin embargo, con K-NN la votación global funciona mejor ya que, al igual que el método Simple, compara más valores que la votación por valor.

Los algoritmos de clustering K-means y EM obtienen resultados muy parecidos y, la mayoría de los mejores se obtienen con el clasificador K-NN.

Observando los espacios de color, se puede comprobar que los tres obtienen resultados muy parejos, aunque, en general, HSV parece obtener algunos mejores y CIE-Lab algunos peores. Aún así, los tres coinciden en que obtienen mejores resultados con el clasificador K-NN.

Finalmente la combinación de parámetros seleccionada para la aplicación, ha sido la que utiliza el espacio de color CIELab, el algoritmo de clustering EM, el algoritmo de clasificación K-NN, el método de clasificación Simple y la votación global, ya que es la que más Accuracy ha obtenido, con un 54.75%.

TABLA 1  
COMBINACIONES DE LOS MÉTODOS

Espacio	Cluster	Clasificador	Método	Votación	Accuracy
LAB	EM	KNN	Simple	global	54.75
RGB	EM	KNN	Simple	global	52.75
LAB	kmeans	KNN	Simple	global	51.87
HSV	kmeans	KNN	Complejo	global	51.75
HSV	EM	KNN	Simple	valor	51.75
HSV	kmeans	KNN	Simple	global	51.12
RGB	kmeans	KNN	Simple	global	50.87
HSV	EM	KNN	Complejo	global	50.75
RGB	kmeans	KNN	Complejo	global	50.37
HSV	EM	KNN	Simple	global	50.25
RGB	EM	KNN	Complejo	global	50.12
HSV	kmeans	KNN	Simple	valor	48.87
RGB	EM	SVM	Complejo	global	47.25
HSV	kmeans	KNN	Complejo	valor	46.37
HSV	kmeans	SVM	Simple	global	46.25
HSV	EM	KNN	Complejo	valor	46.12
HSV	kmeans	SVM	Simple	valor	45.75
HSV	EM	SVM	Simple	valor	45.37
HSV	EM	SVM	Simple	global	45.37
HSV	kmeans	SVM	Complejo	valor	45.25
RGB	kmeans	SVM	Simple	global	44.62
RGB	kmeans	KNN	Simple	valor	44.62
LAB	EM	KNN	Complejo	global	44.62
RGB	EM	SVM	Simple	valor	44.25
RGB	EM	SVM	Simple	global	44.12
RGB	EM	KNN	Simple	valor	44.00
RGB	kmeans	SVM	Complejo	global	43.75
LAB	kmeans	KNN	Complejo	global	43.75
RGB	kmeans	SVM	Simple	valor	43.62
RGB	kmeans	KNN	Complejo	valor	43.62
RGB	EM	SVM	Complejo	valor	43.50
HSV	EM	SVM	Complejo	global	42.50
RGB	kmeans	SVM	Complejo	valor	42.12
LAB	EM	SVM	Complejo	global	41.62
HSV	kmeans	SVM	Complejo	global	40.75
RGB	EM	KNN	Complejo	valor	40.50
LAB	EM	SVM	Complejo	valor	40.25
LAB	EM	KNN	Simple	valor	40.12
HSV	EM	SVM	Complejo	valor	40.12
LAB	kmeans	KNN	Simple	valor	39.87
LAB	EM	KNN	Complejo	valor	39.75
LAB	kmeans	SVM	Complejo	global	39.50
LAB	kmeans	KNN	Complejo	valor	39.50
LAB	kmeans	SVM	Complejo	valor	37.62
LAB	kmeans	SVM	Simple	valor	35.12
LAB	kmeans	SVM	Simple	global	34.87
LAB	EM	SVM	Simple	valor	30.12
LAB	EM	SVM	Simple	global	29.37

Cálculo de las Accuracys de todas las combinaciones de métodos posibles ordenados de mayor a menor.



## 7.2. TIEMPOS DE EJECUCIÓN

En cada una de las implementaciones de los algoritmos, se han comprobado sus respectivos tiempos de ejecución de la misma manera que en el apartado anterior, llegando a la conclusión de que los algoritmos de clustering son los que acaparan la mayoría del tiempo de la ejecución del detector. En la Tabla 2 se muestran los resultados obtenidos, con una CPU i7 a una frecuencia de 3,5 GHz, al utilizar cada espacio de color combinado con los algoritmos de clustering, detallando métricas de tiempo de ejecución.

TABLA 2  
MEDIA DE LOS TIEMPOS DE EJECUCIÓN

	RGB	LAB	HSV
K-means	0.11	0.10	0.08
EM	3.54	2.77	3.45

Tiempos medidos en segundos de las ejecuciones de los algoritmos K-means y EM con los espacios de color RGB, CIELab y HSV.

Como se puede observar en la Tabla 2, en términos de ejecución K-means supera en eficiencia a EM. El tiempo de ejecución de K-means es entre 27 y 43 veces más rápido que el de EM.

En el apartado anterior, se eligió la combinación de parámetros con la mejor Accuracy y esta incluía el uso de EM. Sin embargo, en caso de que la aplicación tuviera que utilizarla un usuario corriente, se debería tener en cuenta que la clasificación de las imágenes fuera lo más rápido posible, eligiendo así la combinación de parámetros con la mejor Accuracy que utilizará K-means.

## 7.3. RESULTADOS DEL MODELO FINAL

A continuación, en la Tabla 3, se muestran los resultados de todas las detecciones realizadas con el modelo final con una matriz de confusión.

TABLA 3  
COMBINACIONES DE LOS MÉTODOS

	Cool	Earthy	Playful	Romantic	Sensual	Serene	Spiritual	Warm
Cool	73.0	0.0	0.0	0.0	0.0	17.0	10.0	0.0
Earthy	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0
Playful	0.0	49.0	29.0	0.0	1.0	10.0	11.0	0.0
Romantic	0.0	13.0	0.0	49.0	7.0	0.0	0.0	31.0
Sensual	0.0	39.0	0.0	0.0	36.0	0.0	25.0	0.0
Serene	10.0	10.0	10.0	0.0	0.0	50.0	10.0	10.0
Spiritual	4.0	0.0	0.0	0.0	20.0	2.0	74.0	0.0
Warm	10.0	50.0	0.0	0.0	0.0	13.0	0.0	27.0

Matriz de confusión de las detecciones con el mejor modelo para todas las emociones. En rojo las emociones en las que la mayoría

Como se observa en la Tabla 3, los resultados de las clasificaciones con las emociones *Cool*, *Earthy*, *Romantic*, *Serene*, *Spiritual* han sido buenos en su mayoría, entre un 49% y un 100% de acierto. En cambio, para las emociones *Playful*, *Sensual* y *Warm* no han dado tan buenos resultados, entre un 27% y un 36% y la mayoría de las imágenes han sido clasificadas como *Earthy*. Esto se debe a que algunos de los conjuntos de colores que definen las emociones se parecen, cómo se puede ver en la Figura 6.

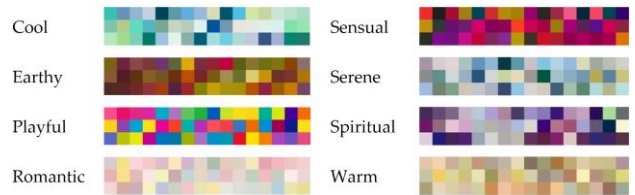


Figura 6. Conjuntos de colores de cada una de las emociones.

Los resultados de algunas clasificaciones, utilizando el modelo final, se pueden ver de una manera más visual comparando los colores de la Figura 6 con los resultados expuestos en las Figuras 7, 8 y 9.



Figura 7. Ejemplos de detecciones correctas (fila 1) e incorrectas (fila 2) para imágenes clasificadas con la emoción *Cool*. Debajo de cada imagen se muestran los colores predominantes de ella y se indica la predicción del mejor modelo.

En la Figura 7, mirando las imágenes mal clasificadas, se puede observar cómo se confunde la emoción *Cool* con la emoción *Serene*. Esto se debe a que los colores de ambas emociones tienen tonos azules y claros que pueden dar a equivocación.



Figura 8. Ejemplos de detecciones correctas (fila 1) e incorrectas (fila 2) para imágenes clasificadas con la emoción *Playful*. Debajo de cada imagen se muestran sus colores predominantes y se indica el resultado que ha obtenido el mejor modelo.

En cambio, en la Figura 8, se está confundiendo la emoción *Playful* con las emociones *Earthy* y *Serene*. La primera se debe a que las dos gamas de colores tienen tonos cálidos similares, cómo el marrón. Sin embargo, la segunda, con la emoción *Serene*, se debe a que gran parte de la imagen tiene tonos azules propios de esa emoción.



Figura 9. Ejemplos de detecciones correctas (fila 1) e incorrectas (fila 2) para imágenes clasificadas con la emoción *Spiritual*. Debajo de cada imagen se muestran los colores predominantes de ella y se indica la predicción del mejor modelo.

Finalmente, en la Figura 9, las detecciones erróneas son con las emociones *Sensual* y *Cool*, ya que los colores que definen la emoción *Spiritual* tiene tonos morados y azules propios de las emociones *Sensual* y *Cool* respectivamente.

## 8 DISEÑO DE LA APLICACIÓN

Como paso final del proyecto, se ha diseñado e implementado una interfaz sencilla con la que poder probar el trabajo desarrollado durante todo el proyecto.

El diseño para la interfaz de la aplicación se ha desarrollado con la ayuda de la herramienta visual QtDesigner, incluida en la librería PyQt4 [16]. Con esta herramienta se ha creado toda la parte visual de la interfaz cómo se puede ver en la Figura 10.

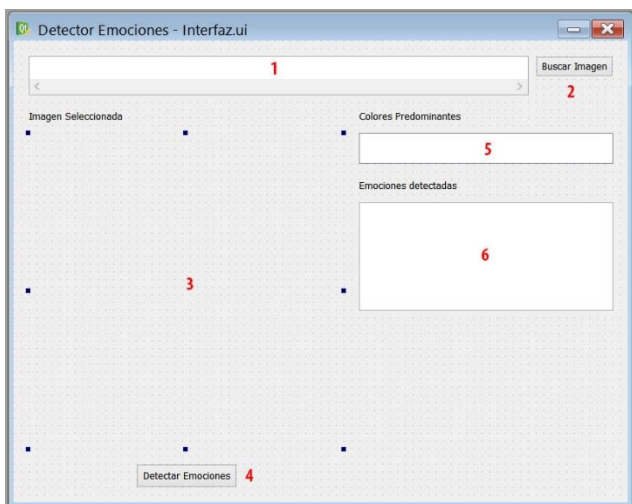


Figura 11. Interfaz de la aplicación desarrollada con QtDesigner.

En la Figura 10 se puede observar cómo los elementos funcionales de la aplicación son los siguientes:

1. Campo de texto superior de la ruta.
2. Botón *Buscar Imagen*.
3. Cuadro de vista de la imagen.
4. Botón *Detectar Emociones*.
5. Cuadro de vista de los colores Predominantes.
6. Campo de texto derecho de la emoción detectada.

Una vez acabada la parte gráfica, se ha creado una clase Python que conecta el diseño, guardado en un archivo *.ui*, con la parte lógica implementada durante todo el proyecto.

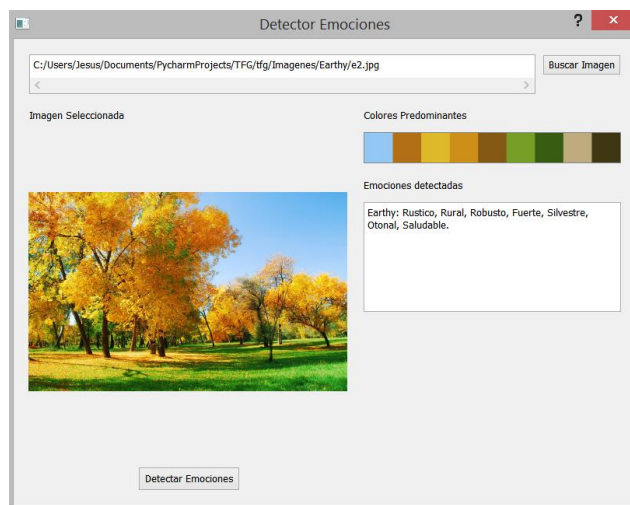


Figura 11. Detección de las emociones transmitidas por una imagen utilizando la aplicación final.

Cómo se puede ver en la Figura 11, se puede buscar la imagen con el campo de texto superior o pulsando el botón *Buscar Imagen*. Acto seguido se mostrará la imagen justo debajo de la etiqueta *Imagen Seleccionada*. Una vez cargada, ya se pueden detectar las emociones que transmite pulsando sobre el botón *Detectar Emociones*. Cuando la aplicación haya obtenido las emociones, se mostraran por una parte, los colores que definen la imagen justo debajo de la etiqueta *Colores Predominantes* y, por otra, la emoción detectada en el campo de texto situada debajo de la etiqueta *Emociones detectadas*.

## 9 CONCLUSIONES Y VÍAS DE CONTINUACIÓN

A lo largo del proyecto se ha visto el proceso que se ha llevado a cabo para desarrollar una aplicación capaz de detectar las emociones que transmite una imagen a partir de los colores que la componen.

Antes de exponer las conclusiones, cabe destacar que gran parte del éxito obtenido ha sido gracias a la utilización del libro [8] que ha proporcionado unos conjuntos de colores etiquetados con emociones, basados en estudios psicológicos que relacionan la combinación de colores con las emociones.

### 9.1. CONCLUSIONES DE LOS RESULTADOS FINALES

Para llevar a cabo la aplicación, se han estudiado, implementado y comparado los espacios de color RGB, CIELab y HSV dónde se representan los colores de la imagen. Los tres han tenido una eficacia muy parecida, aunque HSV ha obtenido unos resultados ligeramente superiores. La distribución de colores que proporciona el valor de la intensidad en HSV ha sido la causante de estos resultados, ya que los colores que definen las emociones se caracterizan por tener intensidades similares.



Para poder representar debidamente los colores de la imagen, se han estudiado, implementado y comparado los algoritmos de clustering K-means y EM para poder representar en unos pocos valores toda la gama cromática. Ambos resultan igual de eficaces, aunque EM consigue los mejores resultados, ya que son ligeramente superiores que los de K-means. A pesar de obtener mejores resultados, EM resulta ser unas 30 veces más lento que K-means y, para una aplicación final en la que tengan que trabajar usuarios diariamente, puede llegar a tener unos tiempos de respuesta demasiado elevados.

Al igual que los algoritmos anteriores, se han estudiado, implementado y comparado los algoritmos de clasificación K-NN y SVM con el fin de etiquetar correctamente los colores predominantes obtenidos de los algoritmos de clustering. Gracias a los datos proporcionados en [8] y a los algoritmos de clasificación, se han podido entrenar modelos de clasificación que sirven de base para el etiquetado de nuevos valores. En este caso, el algoritmo K-NN ha obtenido, con una amplia mayoría, las mejores resultados a diferencia del algoritmo SVM. El motivo es que el K-NN implementado obtiene 9 clasificaciones por cada color predominantes contra 1 clasificación por color en SVM, dotando al primer algoritmo de más precisión. Además, para mejorar los resultados con SVM debería hacerse una búsqueda exhaustiva de parámetros, ya que una de las características de este algoritmo es que puede tener numerosas configuraciones creando diferentes clasificadores.

Finalmente, se han diseñado e implementado diferentes formas de entrenar los clasificadores y de obtener la emoción final que defina la imagen. En cualquier caso, los mejores resultados son los que utilizan conjuntos de datos con mayor número de muestras.

## 9.2. VÍAS DE CONTINUACIÓN

Para aumentar la calidad de la aplicación se podrían probar a mejorar algunas de las implementaciones cómo por ejemplo:

- Utilizar otras librerías científicas para la implementación de los algoritmos.
- Normalizar los datos con diferentes fórmulas y comparar los rendimientos.
- Hacer una búsqueda exhaustiva de los mejores parámetros para el algoritmo SVM.

Además, se podrían añadir algunas funcionalidades nuevas que interactúen con el usuario cómo por ejemplo:

- Mostrar que porcentaje de la emoción detectada transmite la imagen.
- Añadir las emociones secundarias que transmite la imagen.
- Incluir una funcionalidad que proporcione al usuario la posibilidad de añadir colores a las paletas de colores de las emociones.

## AGRADECIMIENTOS

Quiero agradecer a Robert Benavente, el que ha sido mi tutor del Trabajo Final de Grado, por orientarme, corregirme y resolverme cualquier tipo de duda durante el transcurso de todo el proyecto.

También quiero dar las gracias a mis compañeros del grado, por ayudarme y darme consejos con la parte técnica del proyecto.

## BIBLIOGRAFÍA

- [1] IllusionStudio, <http://www.psicologiadelcolor.es> (Junio 2015)
- [2] M. Solli and R. Lenz, "Color Emotions for Image Classification and Retrieval", Proc. IS&Ts 4th European Conference on Colour in Graphics, Imaging, and Vision, CGIV 2008, 367-371, 2008.
- [3] Y. Shing, N. Y. Kim, Y. Kim, E. Y. Kim, "Emotion recognition using color and pattern in textile images", 2008 IEEE Conference on Cybernetics and Intelligent Systems, 1100 – 1105, 2008.
- [4] P. S. Chaitanya, K. V. Narasimha Reddy, G. Madhavi, "Effective Search of Color-Spatial Image Using Semantic Indexing", International Journal of Computer Science, Engineering and Applications (IJCSA) Vol. 2, No. 4, 9-19, 2012
- [5] S. Kobayashi. "Color Image Scale". Kodansha Intern., 1991.
- [6] G. Csurka, S. Skaff, L. Marchesotti, C. Saunders, "Building look & feel concept models from color combinations - With applications in image classification, retrieval, and color transfer", The Visual Computer, Vol. 27, No. 12, 1039-1053, 2011.
- [7] N. Murray, S. Skaff, L. Marchesotti, F. Perronnin, "Toward automatic and flexible concept transfer, Computers and Graphics", Special Section on CANS, Vol.36 No. 6, 622-634, 2012.
- [8] L. Eiseman, "Pantone guide to communicating with color". Grafix Press, Ltd., 2000.
- [9] R. Duda, P. Hart, D. Stork. "Pattern Classification", second edition, 2000.
- [10] Python Software Foundation, Python 2.7, <https://www.python.org> (Junio 2015)
- [11] Itseez, OpenCV, <http://opencv.org/documentation.html> (Junio 2015).
- [12] Numpy developers, NumPy 1.9, <http://www.numpy.org> (Junio 2015).
- [13] Wyszecki, "Color Science: Concepts and Methods, Quantitative Data and Formulae". John Wiley and sons, New York, 2nd edition. Reprinted in 2000.
- [14] Pantone, <http://www.pantone-colours.com> (Junio 2015).
- [15] Itseez, OpenCV SVM, [http://docs.opencv.org/modules/ml/doc/support\\_vector\\_machines.html](http://docs.opencv.org/modules/ml/doc/support_vector_machines.html) (Junio 2015)
- [16] Riverbank Computing Limited, PyQt4, <http://www.riverbankcomputing.co.uk/software/pyqt> (Junio 2015).